UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Coal-seismic, desktop computer programs in BASIC; part 6: develop rms velocity functions and apply mute and normal moveout

by

Wilfred P. Hasbrouck[1]

Open-File Report 83-341

[1] Golden, Colorado

1983

ABSTRACT

Processing of data taken with the U.S. Geological Survey's coal-seismic system is done with a desktop, stand-alone computer. Programs for this computer are written in the extended BASIC language utilized by the Tektronix 4051 Graphic System. This report presents computer programs used to develop rms velocity functions and apply mute and normal moveout to a 12-trace seismogram.

INTRODUCTION

The purpose of this report is to present computer programs used to develop rms (root-mean-square) velocity functions and apply mute and normal moveout (NMO) to a 12-trace seismic record. These data processing procedures are well known in the seismic data processing industry, and the topics are discussed in most elementary exploration geophysics texts. The computer programs of this report were developed as part of the U.S. Geological Survey's coal-seismic system. All computer programs were written in an extended BASIC language developed by Tektronix, Inc. for use with their 4051 Graphic System. The programs require four pieces of Tektronix equipment: a 4051 Graphic System with a 32K-byte memory, a 4924 digital cartridge tape drive, a 4631 Hard Copy unit, and a Data Processing ROM. In addition, a special ROM (discussed in the last section of this report) is used.

All programs are self-prompting. In tracing through a sample problem you will notice that the programs print requests and questions followed by a flashing question mark. The computer then waits for you to enter a response from the keyboard. Replies entered in order to run the sample problems are enclosed in boxes on the figures of this report.

RMS VELOCITY FUNCTIONS

The rms velocity functions developed by the first program of this report are used primarily in the computation and removal of normal moveout, the third program of this report. Rather than being mathematical expressions in closed form, the velocity functions produced by the program are sets of rms velocites matched to sets of reflection times. Velocities used in the development of the functions are obtained from a discrete-layer model, a downhole velocity survey, or a set of X-square/T-square analyses. Examples illustrating the use of the program to develop rms velocity functions from these three sources follow.

The first two blocks shown on the screen displays are the same for all three sample problems. After you enter the date on which the computations were made, you are requested: "ENTER RECORD AND SAMPLE INTERVAL INFORMATION". Here you are to supply the record's beginning, end, and sample-interval time. In sample problem 1, these times are 200, 450, and 2 millisec (msec), respectively. Next you are prompted: "SELECT VELOCITY FUNCTION SOURCE". The selection is made by entering a number from the list.

Using velocities from a discrete-layer model--Sample Problem 1:

Figure 1 shows the screen display and the type of information

```
COMPUTE AND STORE RMS VELOCITY FUNCTION
Computation date (9 char,max): [04 FEB 83]

ENTER RECORD AND SAMPLE INTERVAL INFORMATION
 Time at start of computation = [200]
   Time at end of computation = [450]
  Sample int. for computation = [2]
No of points in vel. function = 126

SELECT VELOCITY FUNCTION SOURCE
   1. Velocity from discrete-layer model
   2. Velocity from downhole velocity survey
   3. Velocity from set of X↑2-T↑2 analyses
ENTER SELECTION FROM ABOVE LIST; NUMBER = [1]

ENTER MODEL PARAMETERS
Model designation(10 char,max): MODEL: TEST
    Number of layers in model = [4]
      VALUES TO BASE OF LAYER
LAYER   REFL TIME   INT VEL    RMS VEL
  1       [160        0.25]     0.250
  2       [240        0.4 ]     0.308
  3       [380        0.6 ]     0.439
  4       [420        0.5 ]     0.445

Interval velocity beneath layer 5 = [1.0]
  5         450         1         0.502

DO YOU WANT TO PLOT VELOCITY FUNCTION? (Y OR N) [Y]
```

Figure 1. Copy of screen display for sample problem 1.

entered (enclosed in boxes) in order to compute a rms velocity function from a set of discrete layers of constant interval velocity. After the interval velocity and reflection time to the base of each layer are entered, the program uses the well-known equation (for example, see Telford and others, 1976, equation 4.41b, p. 271) to compute the rms velocity. If the time at the end of computation (450 msec in this example) is greater than the reflection time to the base of the last layer (420 msec), you are prompted to supply an interval velocity (in the example, 1 km/sec) for a layer beneath the last layer of the model.

Upon electing to plot the velocity function (last line on fig. 1), you are requested to enter plot ranges and tickmark intervals after which a plot is made, such as that shown on figure 2.

```
SET PLOT LIMITS
    Minimum rms velocity = 0.29
    Maximum rms velocity = 0.50
   Min plot rms velocity =[0.2]
   Max plot rms velocity =[0.6]
   rms vel tickmark int. =[0.1]

Minimum-reflection time = 200
Maximum reflection time = 450
Refl time tickmark int. =[50]
```

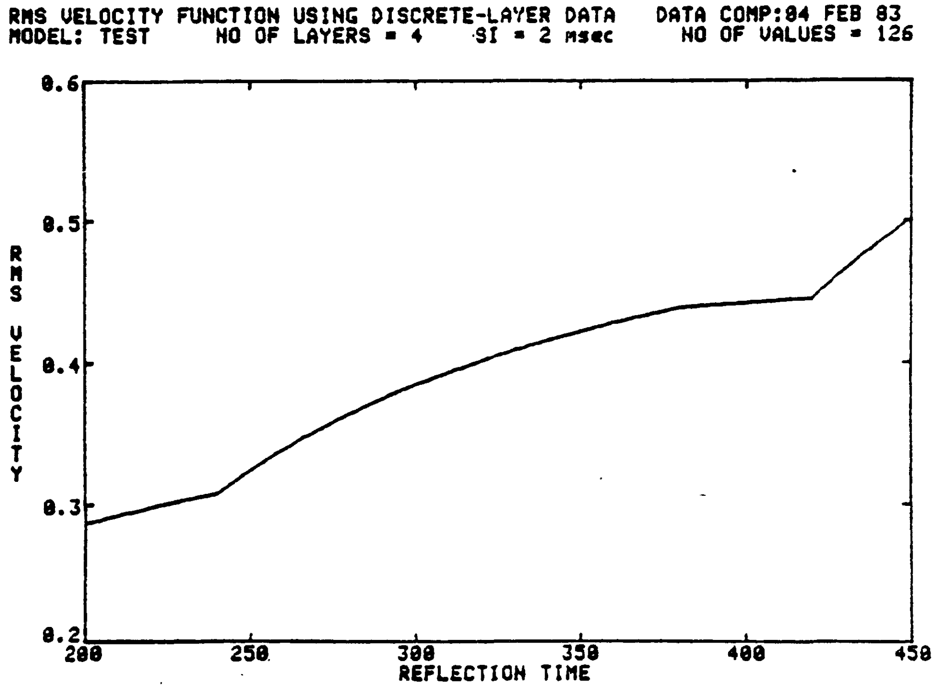

Figure 2. Copy of screen display showing plot limits and plot of velocity function for the four-layer model of sample problem 1.

If you elect to store the time-velocity data, the screen display of figure 3 is produced. To verify that the data have been stored, the program first erases the data from memory and then reads them back into memory from the tape.

```
DO YOU WANT TO STORE VELOCITY FUNCTION? (Y OR N) [Y]
INSERT MDT IN 4924      FILE NO = [55]
ARE YOU READY TO PROCEED? (Y OR N) [Y]
LENGTH OF FILE REQUIRED = 2620
IS FILE OF SUFFICIENT LENGTH? (Y OR N) [N]
INSERT MDT IN 4051
ARE YOU READY TO PROCEED? (Y OR N) [Y]
RE-INSERT MDT IN 4924
ARE YOU READY TO PROCEED? (Y OR N) [Y]
DATA STORED AND READABLE IN FILE 55
PROGRAM COMPLETED
```

Figure 3. Copy of screen display for storing data.

Velocities from a downhole survey--Sample Problem 2:

Figure 4 shows the screen display and the information entered in order to compute a rms velocity function using rms velocities obtained from a downhole survey. After entering information about the survey, you are next prompted to enter one-way times and rms velocities as obtained in the survey. For the sample problem , the last survey observation was at a one-way time of 92.48 msec. Because the reflection time at the end of computation was selected to be 200 msec, a one-way time of 100, you are asked if you want to enter an estimated rms velocity at the one-way time of 100 msec. If you answer with an N then the program will assume that the last rms velocity entered (1.71 km/s) exists over the interval from 92.48 to 100 msec.

Figure 5 is a copy of the screen display of the plotted velocity function for sample problem 2.

Velocities from X-square/T-square analyses--Sample Problem 3:

If velocity had been determined from a set of X-square/T-square analyses, then you would enter a 3 in response to the selection prompt, and a screen display similar to that of figure 6 would be produced. In this example, data from three analyses have been entered. Note that you are asked to supply bounding rms velocities--in this case, 0.2 and 0.8 km/sec at the start and end times of 50 and 450 msec respectively. After electing to make a plot of the velocity function and upon entering the plot limits as shown on the top of figure 7 , the plot shown on lower part of the figure will be produced. Storage of these results proceeds as previously described.

```
COMPUTE AND STORE RMS VELOCITY FUNCTION

ENTER RECORD AND SAMPLE INTERVAL INFORMATION
 Time at start of computation = 40
   Time at end of computation = 200
  Sample int. for computation = 1
No of points in vel. function = 161

SELECT VELOCITY FUNCTION SOURCE
   1. Velocity from discrete-layer model
   2. Velocity from downhole velocity survey
   3. Velocity from set of X↑2-T↑2 analyses
ENTER SELECTION FROM ABOVE LIST; NUMBER = 2

ENTER INFORMATION AND DATA FROM A DOWNHOLE SURVEY
     Survey area (10 char,max) : WATKINS,CO
Hole designation (7 char,max) : 79-101A
      Data date (12 char,max) : 11 FEB 1980

INPUT 1-WAY TIMES AND RMS VELOCITIES
No of observations = 31
OBSERV NO    1-WAY TIME   RMS VEL
    1          27.67        .36
    2          34.87        .41
    3          41.57        .44
    4          44.78        .54
    5          47.19        .65
    6          49.48        .74
    7          52.01        .80
    8          54.30        .86
    9          56.37        .92
    10          58.42       .98
    11          60.36       1.03
    12          62.30       1.08
    13          64.32       1.12
    14          65.65       1.18
    15          67.17       1.24
    16          68.78       1.28
    17          70.50       1.32
    18          72.21       1.35
    19          73.72       1.39
    20          75.23       1.42
    21          76.84       1.45
    22          79.24       1.46
    23          81.35       1.47
    24          83.25       1.49
    25          84.96       1.51
    26          86.36       1.54
    27          87.77       1.57
    28          89.07       1.61
    29          90.28       1.64
    30          91.38       1.68
    31          92.48       1.71

DO YOU WANT TO ESTIMATE RMS VEL AT LAST TIME ON RECORD? (Y OR N) Y
Estimated rms vel at 1-way time of 100 = 1.80

DO YOU WANT TO PLOT VELOCITY FUNCTION? (Y OR N) Y
```

Figure 4. Copy of screen display showing entries for sample problem 2.

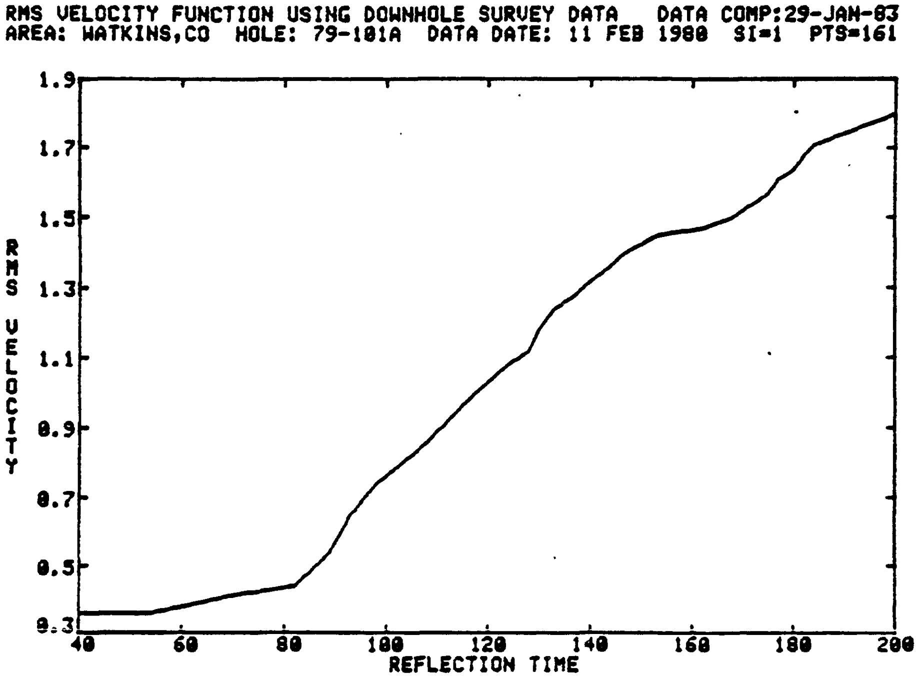

Figure 5. Copy of screen display showing results for sample problem 2.

```
COMPUTE AND STORE RMS VELOCITY FUNCTION

ENTER RECORD AND SAMPLE INTERVAL INFORMATION
 Time at start of computation = [50]
   Time at end of computation = [450]
  Sample int. for computation = [2]
No of points in vel. function = 201

SELECT VELOCITY FUNCTION SOURCE
   1. Velocity from discrete-layer model
   2. Velocity from downhole velocity survey
   3. Velocity from set of X↑2-T↑2 analyses
ENTER SELECTION FROM ABOVE LIST; NUMBER = [3]

ENTER INFORMATION AND DATA FROM X↑2-T↑2 ANALYSES
  Survey area (10 char,max) : [1234567890]
SP designation (7 char,max) : [1234567]
    Data date (12 char,max) : [123456789012]

 Number of X↑2-T↑2 analyses = [3]

COINCIDENT-RAY TIME      X↑2-T↑2 VEL
         [100]                [0.25]
         [300]                [0.5]
         [400]                [0.75]

Est. rms vel at start time of 50 = [0.2]
  Est. rms vel at end time of 450 = [0.8]

DO YOU WANT TO PLOT VELOCITY FUNCTION? (Y OR N) [Y]
```

Figure 6. Copy of screen display showing entries for sample problem 3.

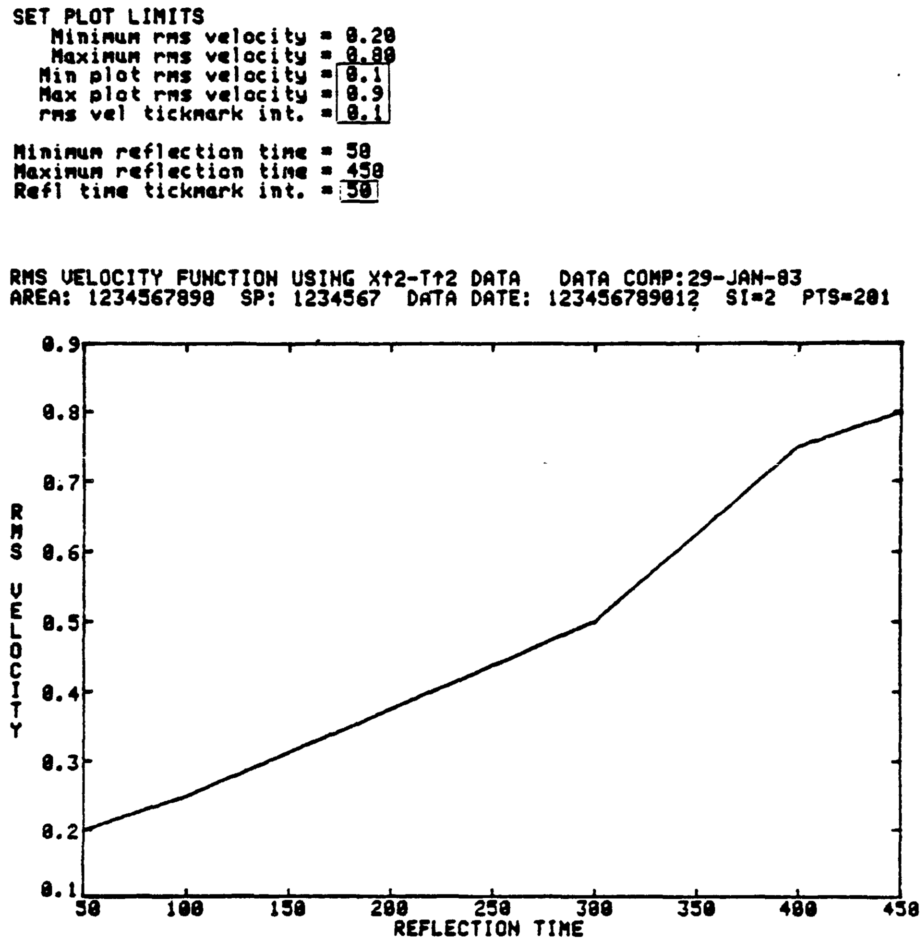

Figure 7. Copy of screen display showing results of sample problem 3.

MUTE AND NORMAL MOVEOUT

The function of the mute program is to suppress high amplitude events associated with the first arrivals; the purpose of the normal moveout programs is to remove normal moveout (NMO) from the reflected arrivals. Both procedures are standard in seismic data processing, (Telford and others, 1976, mute: p. 392, NMO: p. 263). In the sections which follow, sample problems are given to illustrate the use of the muting function and the removal of NMO using both constant and variable velocity functions.

Muting is accomplished in the programs of this report by use of a cosine taper (a multiplier in the form of an inverted, shifted, half-period cosine ranging from 0 to 1) whose length and start time are entered from the keyboard. For a selected spread segment (traces 1 and 12 on a single-ended spread, for example), you are asked to enter the beginning time of the mute for only the first and last traces. Using a linear interpolation, the program then computes mute start times for the interior traces. You are also offered the option of selecting mute-start times individually for each trace. All trace amplitudes prior to the start of the mute are forced to zero.

In shallow-reflection studies, the selection of the start and duration times of the mute function must be done carefully. The objective of the mute is to decrease the amplitude of the first-arrival wave train without seriously diminishing the amplitude of the shallow reflections, a task easier to state than to accomplish. Because NMO removal is a time consuming operation, it is suggested that you set the "end NMO time" to a shortened value during the search for an acceptable mute function. If you are interested only in removing normal moveout within a time window bracketing deeper arrivals, then the mute function need not be applied.

Application od muting to NMO-corrected record--Sample Problem 4:

The record for this example is from a wave test conducted in an area in which shallow shear-wave reflections appear to exist. Velocity from a set of X-square/T-square analyses is 0.21 km/sec. Figure 8 is a

```
CONSTANT VELOCITY NMO CORRECTION OF 12-TRACE RECORD

INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924

Name of NMO MDT = NM6UT
Rec# on NMO MDT = 7
Name of OBS MDT = WT2TH
Rec# on OBS MDT = 7

DO YOU WANT TO APPLY MUTE? (Y OR N) Y

 OBS MDT S.I. = 0.5
 NMO MDT S.I. = 1.0
OBS MDT delay = 0
 Constant vel = 0.21

SPREAD TYPES
   1. Single-ended with equally spaced detectors
   2. Split with equally spaced detectors
   3. Spread with unequally spaced detectors
ENTER SELECTION FROM ABOVE LIST; NUMBER = 1

FOR THE SINGLE-ENDED SPREAD WITH EQUALLY SPACED DETECTORS:
 Trace  1 offset = 3
Tr. 1 mute start = 25
 Trace 12 offset = 36
Tr.12 mute start = 198
  Cos-taper time = 50

FOR NMO-CORRECTED RECORD:
Start NMO time = 0
  End NMO time = 250

PROGRAM COMPLETED
```

Figure 8. Copy of screen display showing entries to produce the muted record for sample problem 4.

copy of the screen display produced by the constant-velocity NMO program. Enclosed in boxes is the information entered in order to produce a muted and NMO-corrected record that ranges in time from 0 to 250 msec.

Displayed on figure 9 is the 0-to-250 msec record segment without

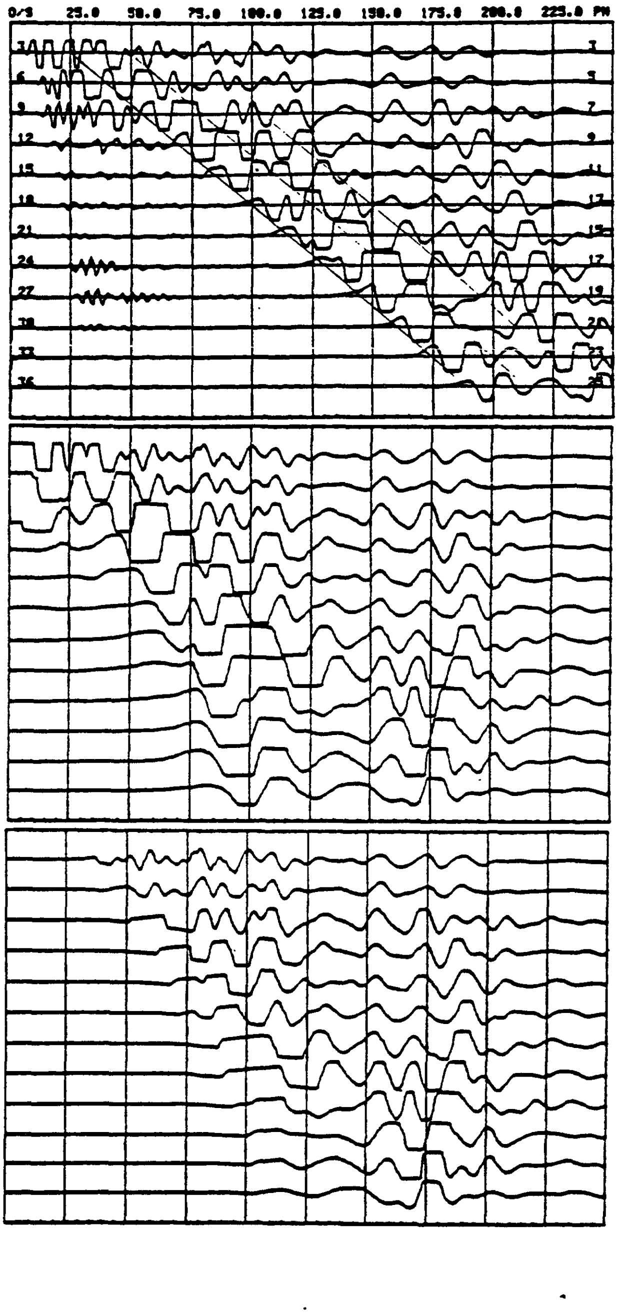

Figure 9. Copy of screen displays showing (upper) record before processing, (middle) record without muting but with NMO applied, and (lower) record with both muting and NMO applied.

any processing (upper), the record segment without mute but with removal of NMO using a constant rms velocity of 0.21 km/sec (middle), and the record (lower) after application of mute using the parameters as entered in figure 8 and after removal of the constant-velocity NMO. Neither static correction nor filtering has been applied to these records. On the unprocessed record, the offset (O/S) in meters from the source point to the detector of each trace is listing in the left column; the position number (PN) of each detector is listing in the right column.

Construction of constant velocity panels--Sample Problem 5:

In this example, the constant-velocity NMO program is used to construct a set of constant-velocity panels at velocities of 0.6, 0.8, and 1.0 km/sec. Figure 10 shows the screen display used to enter data for the unmuted 50-to-250-msec panel at a constant velocity of 0.6 km/sec. Note, as the prompts on figure 8 indicate, that two master data tapes are required when correcting for NMO: one containing the observed data, and one to store the NMO-corrected results. In this example, no muting has been applied.

```
CONSTANT VELOCITY NMO CORRECTION OF 12-TRACE RECORD

INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924

Name of NMO MDT = NM6UT
Rec# on NMO MDT = 3
Name of OBS MDT = AR2UT
Rec# on OBS MDT = 6

DO YOU WANT TO APPLY MUTE? (Y OR N) N

 OBS MDT S.I. = 0.5
 NMO MDT S.I. = 1.0
OBS MDT delay = 0
 Constant vel = 0.6

SPREAD TYPES
   1. Single-ended with equally spaced detectors
   2. Split with equally spaced detectors
   3. Spread with unequally spaced detectors
ENTER SELECTION FROM ABOVE LIST; NUMBER = 1

FOR THE SINGLE-ENDED SPREAD WITH EQUALLY SPACED DETECTORS:
 Trace  1 offset = 36
 Trace 12 offset = 69

FOR NMO-CORRECTED RECORD:
Start NMO time = 50
  End NMO time = 250
PROGRAM COMPLETED
```

Figure 10. Copy of screen display showing entries for 0.6 km/sec panel.

Velocity-panel record segments from 50 to 250 msec with NMO removed at rms velocities of 0.6, 0.8, and 1.0 km/sec are displayed on figure 11. The record segment without NMO correction is shown for comparison.

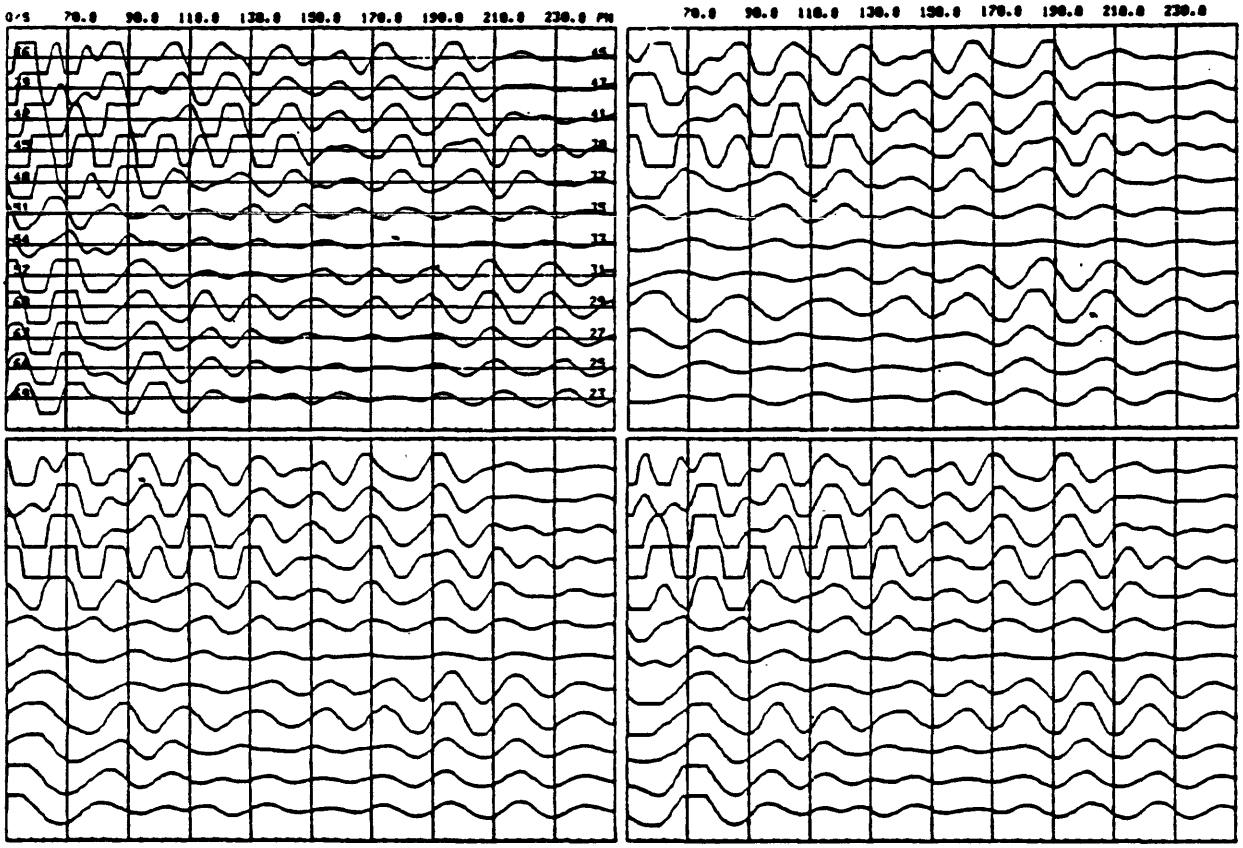

Figure 11. Velocity panels from 50 to 250 msec at velocites in km/sec of 0.6 (upper right), 0.8 (lower left), and 1.0 (lower right). The record segment in the upper-left block is without NMO correction.

Full record with mute and NMO applied--Sample Problem 6:

In this example, the constant-velocity NMO program was used to construct a NMO-corrected record (500 msec) using a velocity of 0.7 km/sec and a 30-msec cosine mute. Muting began on trace 1 at a time of 33 and ended on trace 12 at a time of 47 msec. Figure 12 shows the data-entry screen display to produce the muted and NMO-corrected record shown on the lower half of figure 13.

```
CONSTANT VELOCITY NMO CORRECTION OF 12-TRACE RECORD

INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924

Name of NMO MDT = NM6UT
Rec# on NMO MDT = 10
Name of OBS MDT = AR2UT
Rec# on OBS MDT = 6

DO YOU WANT TO APPLY MUTE? (Y OR N) Y

 OBS MDT S.I. = 0.5
 NMO MDT S.I. = 1.0
OBS MDT delay = 0
 Constant vel = 0.7

SPREAD TYPES
   1. Single-ended with equally spaced detectors
   2. Split with equally spaced detectors
   3. Spread with unequally spaced detectors
ENTER SELECTION FROM ABOVE LIST; NUMBER = 1

FOR THE SINGLE-ENDED SPREAD WITH EQUALLY SPACED DETECTORS:
 Trace  1 offset = 36
Tr. 1 mute start = 33
 Trace 12 offset = 69
Tr.12 mute start = 47
  Cos-taper time = 30

FOR NMO-CORRECTED RECORD:
Start NMO time = 0
  End NMO time = 500
PROGRAM COMPLETED
```

Figure 12. Copy of screen display showing entries for sample problem 6.

Shown on figure 13 are the full records without NMO correction and with normal moveout removed at a constant velocity of 0.7 km/sec. It appears from inspection of the NMO-corrected record that the use of a constant velocity over the entire record may not always be satisfactory.

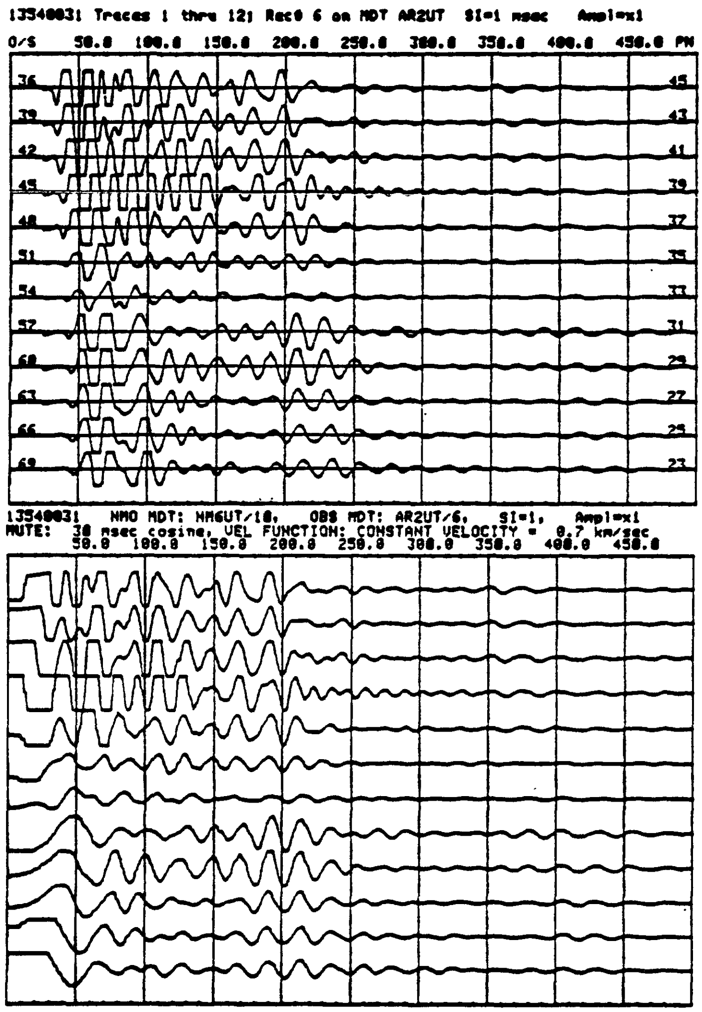

Figure 13. Record 6 on master data tape: AR2UT without NMO (upper) and with a constant 0.7 km/sec NMO velocity and 30-msec cosine mute applied (lower).

NMO correction using a velocity function--Sample Problem 7:

The velocity-function NMO program is illustrated in this last example. Velocities of 1.0 and 0.8 km/sec centered at reflection times of 100 and 200 msec respectively were entered into the X-square/ T-square velocity function program, much as was done in sample problem 3. A start-time velocity of 1.5 and an end-time velocity of 0.6 km/sec were estimated. Start time of computation was 0, end time was 500, and sample interval was 1 msec.

Figure 14 is a copy of the screen display showing the information

```
VELOCITY-FUNCTION NMO CORRECTION OF 12-TRACE RECORD

INSERT VEL-FUNCTION MDT IN 4924   FILE NO = 59

VELOCITY FUNCTION USING X↑2-T↑2 DATA
VERNAL, UT  WT2,W/E  15 MAY 1982
FUNCTION BEGINS AT T=0 AND ENDS AT 500 WITH A SI=1

INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924
Name of NMO MDT = NM6UT
Rec# on NMO MDT = 9
Name of OBS MDT = AR2UT
Rec# on OBS MDT = 6
DO YOU WANT TO APPLY MUTE? (Y OR N) Y

   OBS MDT S.I. = 0.5
  OBS MDT delay = 0

SPREAD TYPES
   1. Single-ended, equally spaced detectors
   2. Split, equally spaced detectors
   3. Spread with unequally spaced detectors
ENTER SELECTION FROM ABOVE LIST; NUMBER = 1

FOR THE SINGLE-ENDED SPREAD:
 Trace  1 offset = 36
Tr. 1 mute start = 33
 Trace 12 offset = 69
Tr.12 mute start = 47
  Cos-taper time = 30

FOR NMO-CORRECTED RECORD:
Start NMO time = 0
  End NMO time = 500
PROGRAM COMPLETED
```

Figure 14. Copy of screen display showing entries for sample problem 7.

entered in order to compute velocity-function NMO correction of the record stored as record 6 on master data tape AR2UT. The NMO-corrected record was stored as record 8 on master data tape NM6UT. Because of the computer's limited memory, a maximum of only 501 velocity-function values, 1001 seismic-trace values, and 130 cosine-taper values are permitted. Also because of the small capacity of the computer, NMO correction is made and stored trace-by-trace and some calculations that should be made outside of a loop are forced to be made within a loop. If this program is to be used on a larger-memory machine, then the program should be modififed to make it run more efficiently.

Shown on figure 15 is a plot of the velocity function (upper) and a display of the record (lower) after velocity-function NMO correction.

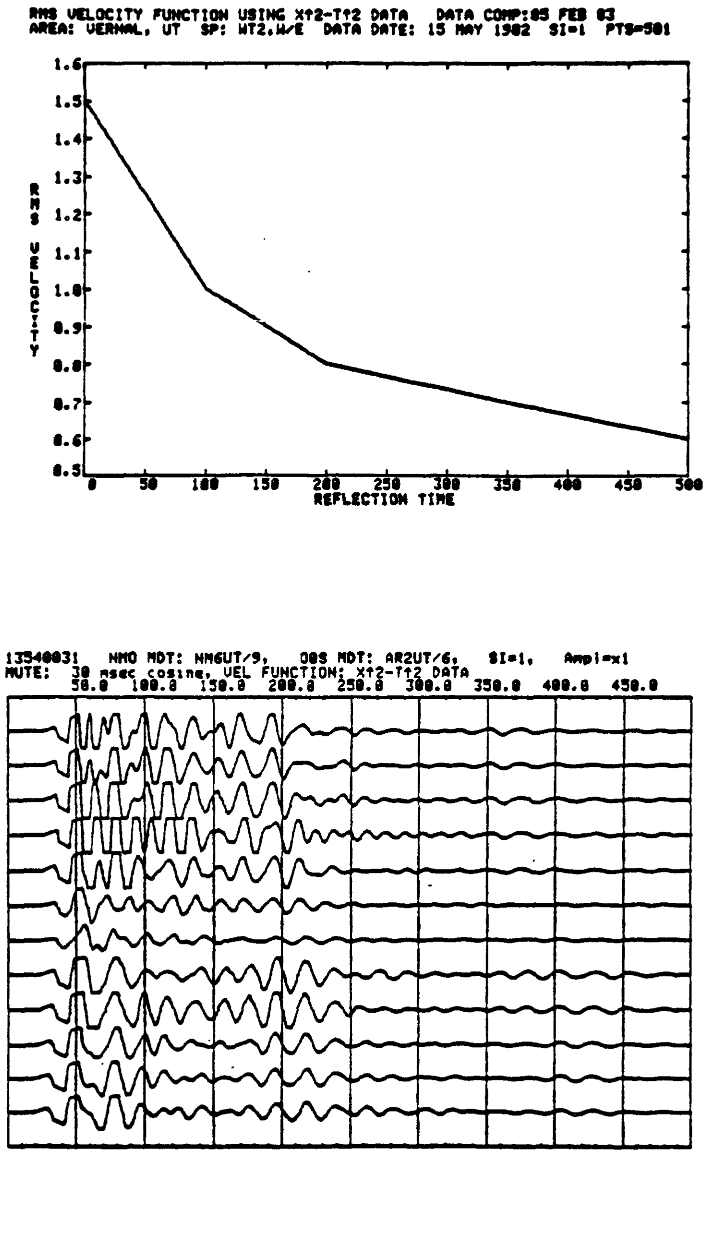

Figure 15. Copy of screen displays showing the plot of the rms velocity function (upper) used in making the velocity-function-NMO corrected record shown on the bottom half of the figure.

COMMENTS ON AND LISTINGS OF THE PROGRAMS

In order to put the programs of this report to work, you must know how to perform the following operations:

1. transcribe the programs into the computer,
2. store the programs on magnetic tape,
3. retreive the programs from magnetic tape,
4. enter information from the keyboard, and
5. copy the screen display.

These tasks are well documented in the computer's operator's manuals.

Four control characters (ones requiring the holding down of the control key as the letter is entered) are used in the programs: G (ring bell), K (move cursor up one line), L (erase screen and move cursor to the HOME position), and the RUB OUT (move cursor to the left margin and down one line). In the printed listing these control characters are shown as G_, K_, L_, and __, respectively.

To achieve maximum data packing on master data tapes, all record data are stored as three-byte hexadecimal values. A specially designed ROM is used to convert MDT (master-data-tape) data from hexadecimal to digital values and later to convert digital values to hexadecimal prior to storing them on a master data tape. The occurrences of these ROMs can be recognized in the programs by the statements of the form: CALL "HEXDEC",B$,V,LEN (B$),3 (where B$ is the string variable containing data in hexadecimal and V is the array containing the data in decimal), and CALL "DECHEX",B$,V,1001,3 (where 1001 is the size of the V array). ROMs also are used to speed the finding of the minimum and maximum values of a function--recognizeable in the program listings by statements of the form: CALL "MIN",V,R1,I1 and CALL "MAX",V,R2,I2G--and to plot a function--identifiable by a statement of the form: CALL "DISP",V. All of the above operations can be performed by BASIC programs; however, it is considerably faster to employ ROMs.

REFERENCE

Telford, W. M., Geldart, L. P., Sheriff, R. E., and Keys, D. A., 1976, Applied geophysics: New York, Cambridge University Press, 860 p.

PROGRAM LISTING FOR GENERATION OF RMS VELOCITY FUNCTIONS

```
100 PRINT "L_COMPUTE AND STORE RMS VELOCITY FUNCTION"
110 INIT
120 DIM A$(12),B$(11),D$(19),G$(1),I$(9),S$(2),U$(9),X1(12)
130 PRINT "Computation date (9 char,max): ";
140 INPUT B$
150 PRINT "__ENTER RECORD AND SAMPLE INTERVAL INFORMATION"
160 PRINT " Time at start of computation = ";
170 INPUT T1
180 PRINT "   Time at end of computation = ";
190 INPUT T2
200 PRINT "  Sample int. for computation = ";
210 INPUT S1
220 N=INT((T2-T1)/S1+1)
230 PRINT "No of points in vel. function = ";N
240 DIM T(N),V(N)
250 T(1)=T1
260 T(N)=T2
270 T4=T1
280 FOR J=2 TO N-1
290 T4=T4+S1
300 T(J)=T4
310 NEXT J
320 PRINT "__SELECT VELOCITY FUNCTION SOURCE"
330 PRINT "   1. Velocity from discrete-layer model"
340 PRINT "   2. Velocity from downhole velocity survey"
350 PRINT "   3. Velocity from set of X↑2-T↑2 analyses"
360 PRINT "ENTER SELECTION FROM ABOVE LIST; NUMBER = ";
370 INPUT N0
380 GO TO N0 OF 390,430,470
390 V$="DISCRETE-LAYER DATA"
400 Q=1
410 GOSUB 690
420 GO TO 500
430 V$="DOWNHOLE SURVEY DATA"
440 Q=2
450 GOSUB 1260
460 GO TO 500
470 V$="X↑2-T↑2 DATA"
480 Q=3
490 GOSUB 1870
500 PRINT "__DO YOU WANT TO PLOT VELOCITY FUNCTION? (Y OR N) ";
510 INPUT G$
520 IF G$="N" THEN 560
530 REM ** PLOT VELOCITY FUNCTION
540 GOSUB 630
550 GOSUB 2500
560 PRINT "__DO YOU WANT TO STORE VELOCITY FUNCTION? (Y OR N) ";
570 INPUT G$
580 IF G$="N" THEN 610
590 REM ** STORE VELOCITY FUNCTION
600 GOSUB 2240
610 PRINT "G_G_G___PROGRAM COMPLETED"
620 END
630 REM ** SUB: DEFAULT DISPLAY AND MOVE TO PAGE BOTTOM
640 WINDOW 0,130,0,100
650 VIEWPORT 0,130,0,100
660 MOVE 0,0
670 PRINT
680 RETURN
690 REM ** SUB: COMPUTE RMS VELOCITIES FROM MODEL
700 PRINT "__ENTER MODEL PARAMETERS"
710 PRINT "Model designation(10 char,max): ";
720 INPUT A$
730 U$=B$
740 I$="SI ="
750 S$=STR(S1)
```

```
760 I$=I$&S$
770 PRINT "    Number of layers in model = ";
780 INPUT N1
790 D$="NO OF LAYERS ="
800 S$=STR(N1)
810 D$=D$&S$
820 N2=N1+1
830 DIM B(N1),C3(N1),J1(N2+1),T0(N2),V0(N2)
840 T0(1)=0
850 B1=0
860 C1=0
870 PRINT "       VALUES TO BASE OF LAYER"
880 PRINT "LAYER   REFL TIME   INT VEL   RMS VEL"
890 IMAGE "K_                          -   ",D.3D
900 FOR J=2 TO N2
910 PRINT "  ";J-1;"         ";
920 INPUT T0(J)
930 PRINT "K_                         ";
940 INPUT V0(J-1)
950 B2=(T0(J)-T0(J-1))/2
960 B1=B1+B2
970 C1=C1+V0(J-1)*V0(J-1)*B2
980 C3(J-1)=C1
990 V1=SQR(C1/B1)
1000 PRINT USING 890:V1
1010 NEXT J
1020 IF T0(J-1)>T2 THEN 1110
1030 PRINT "__Interval velocity beneath layer ";J-1;" = ";
1040 INPUT V0(N2)
1050 B2=(T2-T0(J-1))/2
1060 B1=B1+B2
1070 C1=C1+V0(N2)*V0(N2)*B2
1080 V1=SQR(C1/B1)
1090 PRINT "  ";J-1;"         ";T2;"         ";V0(N2)
1100 PRINT USING 890:V1
1110 REM ** DETERMINE RMS VELOCITY FUNCTION
1120 FOR J=1 TO N
1130 IF T(J)>T0(2) THEN 1160
1140 V(J)=V0(1)
1150 GO TO 1230
1160 FOR L=1 TO N1-1
1170 IF T(J)>T0(L+1) AND T(J)<=T0(L+2) THEN 1190
1180 NEXT L
1190 B1=0.5*(T(J)-T0(L+1))
1200 B2=0.5*T(J)
1210 C2=C3(L)+V0(L+1)*V0(L+1)*B1
1220 V(J)=SQR(C2/B2)
1230 NEXT J
1240 DELETE B1,B2,C1,C2,C3
1250 RETURN
1260 REM ** SUB: COMPUTE VELOCITIES FROM DOWNHOLE-SURVEY RESULTS
1270 PRINT "__ENTER INFORMATION AND DATA FROM A DOWNHOLE SURVEY"
1280 PRINT "    Survey area (10 char,max) : ";
1290 INPUT A$
1300 PRINT "Hole designation (7 char,max) : ";
1310 INPUT U$
1320 PRINT "       Data date (12 char,max) : ";
1330 INPUT D$
1340 DELETE J1
1350 J1=T1/S1+1
1360 J2=T2/S1+1
1370 PRINT "__INPUT 1-WAY TIMES AND RMS VELOCITIES"
1380 PRINT "No of observations = ";
1390 INPUT N1
1400 DIM J0(N1+1),T0(N1),V0(N1+1)
1410 J0=0
1420 V0=0
```

```
1430 PRINT "OBSERV NO   1-WAY TIME   RMS VEL"
1440 FOR J=1 TO N1
1450 IF J<11 THEN 1530
1460 GOSUB 630
1470 PRINT "OBSERV NO   1-WAY TIME   RMS VEL"
1480 FOR J=11 TO N1
1490 IF J<41 THEN 1530
1500 GOSUB 630
1510 PRINT "OBSERV NO   1-WAY TIME   RMS VEL"
1520 FOR J=41 TO N1
1530 PRINT "     "; J; "        ";
1540 INPUT T0(J)
1550 J0(J)=INT(2*T0(J)/S1+0.5)
1560 PRINT "K_                         ";
1570 INPUT V0(J)
1580 NEXT J
1590 IF T2<T0(N1) THEN 1690
1600 PRINT "__DO YOU WANT TO ESTIMATE RMS VEL AT LAST TIME ON RECORD?";
1610 PRINT " (Y OR N) ";
1620 INPUT G$
1630 IF G$="N" THEN 1680
1640 PRINT "Estimated rms vel at 1-way time of "; T2/2; " = ";
1650 INPUT V0(N1+1)
1660 J0(N1+1)=J2
1670 GO TO 1690
1680 V0(N1+1)=V0(N1)
1690 REM ** DETERMINE RMS VELOCITY FUNCTION
1700 L=1
1710 IF J0(1)<J1 THEN 1760
1720 N3=J0(1)-J1
1730 FOR L=1 TO N3
1740 V(L)=V0(1)
1750 NEXT L
1760 L=L-1
1770 FOR K=1 TO N1
1780 N2=J0(K+1)-J0(K)
1790 FOR J=1 TO N2
1800 L=L+1
1810 S=(V0(K+1)-V0(K))/N2
1820 V(L)=V0(K)+(J-1)*S
1830 NEXT J
1840 NEXT K
1850 V(L+1)=V0(N1+1)
1860 RETURN
1870 REM ** SUB: COMPUTE VELOCITIES FROM SET OF X↑2-T↑2 ANALYSES
1880 PRINT "__ENTER INFORMATION AND DATA FROM X↑2-T↑2 ANALYSES"
1890 PRINT "  Survey area (10 char, max) : ";
1900 INPUT A$
1910 PRINT "SP designation (7 char, max) : ";
1920 INPUT U$
1930 PRINT "    Data date (12 char, max) : ";
1940 INPUT D$
1950 PRINT "__ Number of X↑2-T↑2 analyses = ";
1960 INPUT N1
1970 N1=2+N1
1980 DIM T3(N1), V3(N1)
1990 PRINT "__COINCIDENT-RAY TIME     X↑2-T↑2 VEL"
2000 FOR J=2 TO N1-1
2010 PRINT "        ";
2020 INPUT T3(J)
2030 PRINT "K_                         ";
2040 INPUT V3(J)
2050 NEXT J
2060 REM ** LINEAR INTERPOLATION OF INTERMEDIATE RMS VALUES
2070 PRINT "__Est. rms vel at start time of "; T1; " = ";
2080 INPUT V3(1)
2090 V(1)=V3(1)
```

```
2100 PRINT "  Est. rms vel at end time of ";T2;" = ";
2110 INPUT V3(N1)
2120 V(N)=V3(N1)
2130 T3(1)=T(1)
2140 T3(N1)=T(N)
2150 FOR J=2 TO.N-1
2160 FOR K=1 TO N1-1
2170 IF T(J)=>T3(K) AND T(J)<=T3(K+1) THEN 2200
2180 NEXT K
2190 GO TO 2220
2200 C=(V3(K+1)-V3(K))/(T3(K+1)-T3(K))
2210 V(J)=C*(T(J)-T3(K))+V3(K)
2220 NEXT J
2230 RETURN
2240 REM ** SUB: STORE RMS VELOCITY FUNCTION
2250 PRINT "G_G_G___INSERT MDT IN 4924      FILE NO = ";
2260 INPUT F0
2270 GOSUB 3380
2280 F1=20*N+100
2290 PRINT "__LENGTH OF FILE REQUIRED = ";F1
2300 PRINT "__IS FILE OF SUFFICIENT LENGTH? (Y OR N) ";
2310 INPUT G$
2320 IF G$="Y" THEN 2390
2330 PRINT "G_G_G___INSERT MDT IN 4051"
2340 GOSUB 3380
2350 FIND F0
2360 MARK 1,F1
2370 PRINT "G_G_G___RE-INSERT MDT IN 4924"
2380 GOSUB 3380
2390 FIND @2:F0
2400 WRITE @2:N,A$,D$,U$,V$,T,V
2410 PRINT @2,2:
2420 REM ** CHECK ON READABILITY OF STORED DATA
2430 DELETE N,A$,D$,U$,V$,T,V
2440 FIND @2:F0
2450 READ @2:N
2460 DIM T(N),V(N)
2470 READ @2:A$,D$,U$,V$,T,V
2480 PRINT "__DATA STORED AND READABLE IN FILE ";F0
2490 RETURN
2500 REM ** SUB: PLOT VELOCITY FUNCTION
2510 DIM R$(14),T$(17)
2520 R$="RMS VELOCITY"
2530 T$="REFLECTION TIME"
2540 RESTORE 2550
2550 DATA 1,10,125,10,90,1.81,2.86
2560 READ B0,B1,B2,C1,C2,K3,K4
2570 B3=B2-B1
2580 C3=C2-C1
2590 PRINT "SET PLOT LIMITS"
2600 CALL "MIN",V,R1,I1
2610 CALL "MAX",V,R2,I2
2620 IMAGE "   Minimum rms velocity = ",D.2D
2630 PRINT USING 2620:R1
2640 IMAGE "   Maximum rms velocity = ",D.2D
2650 PRINT USING 2640:R2
2660 PRINT "  Min plot rms velocity = ";
2670 INPUT R3
2680 PRINT "  Max plot rms velocity = ";
2690 INPUT R4
2700 PRINT "  rms vel tickmark int. = ";
2710 INPUT R5
2720 R6=(R4-R3)/R5
2730 R7=C3/R6
2740 PRINT "__Minimum reflection time = ";T1
2750 PRINT "Maximum reflection time = ";T2
2760 PRINT "Refl time tickmark int. = ";
```

```
2770 INPUT T5
2780 T6=(T2-T1)/T5
2790 T7=B3/T6
2800 REM ** PLOT BORDERS AND LABELS
2810 PRINT "L_RMS VELOCITY FUNCTION USING ";V$;"   DATA COMP:";B$
2820 GO TO Q OF 2830,2850,2880
2830 PRINT A$;"      ";D$;"    ";I$;" msec      NO OF VALUES = ";N
2840 GO TO 2900
2850 PRINT "AREA: ";A$;"  HOLE: ";U$;"  DATA DATE: ";D$;"  SI=";S1;" ";
2860 PRINT " PTS=";N
2870 GO TO 2900
2880 PRINT "AREA: ";A$;"  SP: ";U$;"  DATA DATE: ";D$;"  SI=";S1;" ";
2890 PRINT " PTS=";N
2900 MOVE B1,C2
2910 RDRAW B3,0
2920 RDRAW 0,-C3
2930 RDRAW -B3,0
2940 RDRAW 0,C3
2950 MOVE 0,(C1+C2)/2+K4*(0.5*LEN(R$)-1)
2960 FOR J=1 TO LEN(R$)
2970 L$=SEG(R$,J,1)
2980 PRINT L$
2990 NEXT J
3000 MOVE 0.5*(B1+B2-K3*LEN(T$)),C1-2*K4
3010 PRINT T$
3020 REM ** PLOT AND LABEL TICKMARKS
3030 MOVE B1-3*K3,C1
3040 IMAGE D.D
3050 PRINT USING 3040:R3
3060 MOVE B1,C1
3070 FOR J=1 TO R6
3080 RMOVE 0,R7
3090 RDRAW B0,0
3100 RMOVE B3-2*B0,0
3110 RDRAW B0,0
3120 RMOVE -B3,0
3130 RMOVE -3*K3,-0.5*K4
3140 PRINT USING 3040:R3+J*R5
3150 RMOVE 3*K3,0.5*K4
3160 NEXT J
3170 MOVE B1,C1
3180 IMAGE 4D
3190 RMOVE -2.3*K3,-K4
3200 PRINT USING 3180:T1
3210 MOVE B1,C1
3220 FOR J=1 TO T6
3230 RMOVE T7,0
3240 RDRAW 0,B0
3250 RMOVE 0,C3-2*B0
3260 RDRAW 0,B0
3270 RMOVE 0,-C3
3280 RMOVE -2.3*K3,-K4
3290 PRINT USING 3180:T1+J*T5
3300 RMOVE 2.3*K3,K4
3310 NEXT J
3320 REM ** PLOT VELOCITY FUNCTION
3330 VIEWPORT B1,B2,C1,C2
3340 WINDOW 1,N,R3,R4
3350 CALL "DISP",V
3360 GOSUB 630
3370 RETURN
3380 REM ** SUB: READY TO PROCEED?
3390 PRINT "__ARE YOU READY TO PROCEED? (Y OR N) ";
3400 INPUT G$
3410 IF G$="N" THEN 3390
3420 RETURN
```

PROGRAM LISTING FOR CONSTANT-VELOCITY NORMAL MOVEOUT CORRECTION

```
100 PRINT "L_CONSTANT VELOCITY NMO CORRECTION OF 12-TRACE RECORD"
110 INIT
120 DIM C$(8),G$(1),H$(11),J$(18),L$(2),M$(17),N$(7),O$(7)
130 DIM S$(1),V$(35),X1(12)
140 Q1=1
150 M$="NO MUTE APPLIED"
160 PRINT "G_G_G___INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924"
170 PRINT "__Name of NMO MDT = ";
180 INPUT N$
190 PRINT "Rec# on NMO MDT = ";
200 INPUT R5
210 F0=R5*12-10
220 PRINT "Name of OBS MDT = ";
230 INPUT O$
240 PRINT "Rec# on OBS MDT = ";
250 INPUT R1
260 F1=R1*12-10
270 PRINT "__DO YOU WANT TO APPLY MUTE? (Y OR N) ";
280 INPUT G$
290 IF G$="N" THEN 320
300 DIM J8(12)
310 Q1=2
320 REM ** ESTABLISH SI, DELAY, AND VEL
330 GOSUB 500
340 REM ** ENTER SPREAD DATA AND MUTE-START TIMES
350 GOSUB 1710
360 IF Q1=1 THEN 390
370 PRINT "  Cos-taper time = ";
380 INPUT C1
390 REM ** ESTABLISH NMO TIME RANGE
400 GOSUB 650
410 REM ** TEST DATA LIMITS
420 GOSUB 770
430 REM ** SET COS TAPER & COMPUTE MUTE COEFFICIENTS
440 IF Q1=1 THEN 460
450 GOSUB 900
460 REM ** COMPUTE, APPLY, AND STORE NMO TRACE-BY-TRACE
470 GOSUB 1000
480 PRINT "G_G_G_PROGRAM COMPLETED"
490 END
500 REM ** SUB: ESTABLISH S. I., DELAY TIME, AND VELOCITY
510 PRINT "__ OBS MDT S. I.  = ";
520 INPUT S1
530 PRINT " NMO MDT S. I.  = ";
540 INPUT S2
550 S3=S2/S1
560 PRINT "OBS MDT delay = ";
570 INPUT L
580 J7=L/S1+1
590 PRINT " Constant vel = ";
600 INPUT V
610 C$=STR(V)
620 V$="CONSTANT VELOCITY = "&C$
630 V$=V$&" km/sec"
640 RETURN
650 REM ** SUB: NMO TIME RANGE
660 PRINT "__FOR NMO-CORRECTED RECORD:"
670 PRINT "Start NMO time = ";
680 INPUT T2
690 J2=T2/S1+1
700 PRINT "  End NMO time = ";
710 INPUT T3
720 J3=T3/S1+1
```

```
730 N2=J3-J2+1
740 N3=(N2-1)/S3+1
750 N4=3*N3
760 RETURN
770 REM ** SUB: TEST DATA LIMITS
780 CALL "MIN",X1,X4,I4
790 CALL "MAX",X1,X3,I3
800 T4=SQR(X4/V*X4/V+T2*T2)
810 IF T4=>L THEN 840
820 PRINT "G_G_G_ERROR: INITIAL COINCIDENT RAY TIME TOO SMALL"
830 GO TO 670
840 J4=INT(0.5+(T4-L)/S1)+1
850 T4=(J4-1)*S1
860 T5=SQR(X3/V*X3/V+T3*T3)
870 J5=INT(0.5+(T5-L)/S1)+1
880 T5=(J5-1)*S1
890 RETURN
900 REM ** SUB: SET COS TAPER
910 C$=STR(C1)
920 M$=C$&" msec cosine"
930 C1=INT(C1/S1+0.5)
940 DIM K9(C1)
950 C2=PI/C1
960 FOR N=1 TO C1
970 K9(N)=0.5-0.5*COS(C2*N)
980 NEXT N
990 RETURN
1000 REM ** SUB: COMPUTE,APPLY,AND STORE TRACE-BY-TRACE
1010 FOR J=1 TO 12
1020 DELETE D$,O
1030 J9=1001 MAX J5
1040 DIM D$(3005),O(J9)
1050 IF J>1 THEN 1090
1060 FIND @2:F1
1070 READ @2:H$,D$
1080 GO TO 1110
1090 FIND @2:F1+J-1
1100 READ @2:D$
1110 O=511
1120 CALL "HEXDEC",D$,O,LEN(D$),3
1130 DELETE D$
1140 O=O-511
1150 IF Q1=1 THEN 1240
1160 FOR K=1 TO J8(J)
1170 O(K)=0
1180 NEXT K
1190 P=0
1200 FOR N=K TO K+C1-1
1210 P=P+1
1220 O(N)=K9(P)*O(N)
1230 NEXT N
1240 GOSUB 1290
1250 GOSUB 1460
1260 GOSUB 1560
1270 NEXT J
1280 RETURN
1290 REM ** SUB: COMPUTE NMO TIMES AS INDICES
1300 DIM T(N3)
1310 J6=J2-S3
1320 T6=T2-S2
1330 FOR I=1 TO N3
1340 J6=J6+S3
1350 T6=T6+S2
1360 T(I)=T6
```

```
1270 NEXT I
1280 DIM M(N3)
1290 FOR K=1 TO N3
1400 M(K)=SQR(T(K)*T(K)+X1(J)/V*X1(J)/V)-T(K)
1410 M(K)=INT(0.5+M(K)/S1)
1420 NEXT K
1430 M=M-J7
1440 DELETE T
1450 RETURN
1460 REM ** SUB: MAKE NMO CORRECTIONS
1470 DIM D(N3)
1480 D=0
1490 J6=J2-S3
1500 FOR I=1 TO N3
1510 J6=J6+S3
1520 D(I)=O(J6+M(I))
1530 NEXT I
1540 DELETE M,O
1550 RETURN
1560 REM ** SUB: STORE NMO-CORRECTED DATA.
1570 DELETE D$
1580 DIM D$(N4)
1590 D$=" "
1600 D=D+511
1610 CALL "DECHEX",D$,D,N3,3
1620 DELETE D
1630 IF J>1 THEN 1670
1640 FIND F0
1650 WRITE H$,M$,N$,O$,V$,N3,R1,S2,T2,T3,D$
1660 GO TO 1690
1670 FIND F0+J-1
1680 WRITE D$
1690 CLOSE
1700 RETURN
1710 REM ** SUB: ENTER SPREAD DATA & MUTE START TIME
1720 PRINT "__SPREAD TYPES"
1730 PRINT "   1. Single-ended with equally spaced detectors"
1740 PRINT "   2. Split with equally spaced detectors"
1750 PRINT "   3. Spread with unequally spaced detectors"
1760 PRINT "ENTER SELECTION FROM ABOVE LIST; NUMBER = ";
1770 INPUT N0
1780 GO TO N0 OF 1790,1810,1830
1790 GOSUB 1850
1800 GO TO 1840
1810 GOSUB 2080
1820 GO TO 1840
1830 GOSUB 2490
1840 RETURN
1850 REM ** SUB: SINGLE-ENDED SPREAD WITH EQUALLY SPACED DETECTORS
1860 PRINT "__FOR THE SINGLE-ENDED SPREAD WITH EQUALLY SPACED DETECTORS:"
1870 PRINT " Trace  1 offset = ";
1880 INPUT X1(1)
1890 IF Q1=1 THEN 1920
1900 PRINT "Tr.  1 mute start = ";
1910 INPUT T8
1920 PRINT " Trace 12 offset = ";
1930 INPUT X1(12)
1940 IF Q1=1 THEN 1970
1950 PRINT "Tr. 12 mute start = ";
1960 INPUT T9
1970 X2=(X1(12)-X1(1))/11
1980 IF Q1=1 THEN 2040
1990 T7=(T9-T8)/11
2000 FOR K=1 TO 12
```

```
2010 T0=(K-1)*T7+T8+L
2020 J8(K)=INT(T0/S1+1.5)
2030 NEXT K
2040 FOR J=2 TO 11
2050 X1(J)=X1(J-1)+X2
2060 NEXT J
2070 RETURN
2080 REM ** SUB: SPLIT SPREAD WITH EQUALLY SPACED DETECTORS
2090 PRINT "__FOR THE SPLIT SPREAD WITH EQUALLY SPACED DETECTORS:"
2100 PRINT " Trace 1 offset = ";
2110 INPUT X1(1)
2120 IF Q1=1 THEN 2150
2130 PRINT "Tr.1 mute start = ";
2140 INPUT T8
2150 PRINT " Trace 6 offset = ";
2160 INPUT X1(6)
2170 IF Q1=1 THEN 2250
2180 PRINT "Tr.6 mute start = ";
2190 INPUT T9
2200 T7=(T9-T8)/5
2210 FOR K=1 TO 6
2220 T0=(K-1)*T7+T8+L
2230 J8(K)=INT(T0/S1+1.5)
2240 NEXT K
2250 X2=(X1(6)-X1(1))/5
2260 FOR J=2 TO 5
2270 X1(J)=X1(J-1)+X2
2280 NEXT J
2290 PRINT " Trace 7 offset = ";
2300 INPUT X1(7)
2310 IF Q1=1 THEN 2340
2320 PRINT "Tr.7 mute start = ";
2330 INPUT T8
2340 PRINT "Trace 12 offset = ";
2350 INPUT X1(12)
2360 IF Q1=1 THEN 2440
2370 PRINT "Tr.12 mute start= ";
2380 INPUT T9
2390 T7=(T9-T8)/5
2400 FOR K=7 TO 12
2410 T0=(K-7)*T7+T8+L
2420 J8(K)=INT(T0/S1+1.5)
2430 NEXT K
2440 X3=(X1(12)-X1(7))/5
2450 FOR J=8 TO 11
2460 X1(J)=X1(J-1)+X3
2470 NEXT J
2480 RETURN
2490 REM ** SUB: SPREAD WITH UNEQUALLY SPACED DETECTORS
2500 PRINT "FOR THE SPREAD WITH UNEQUALLY SPACED DETECTORS:"
2510 IF Q1=1 THEN 2540
2520 PRINT "TRACE NO.   OFFSET   MUTE START TIME"
2530 GO TO 2550
2540 PRINT "TRACE NO.   OFFSET"
2550 FOR J=1 TO 12
2560 PRINT "    ";J;"          ";
2570 INPUT X1(J)
2580 IF Q1=1 THEN 2620
2590 PRINT "K_                          ";
2600 INPUT T0
2610 J8(J)=INT(T0/S1+1.5)
2620 NEXT J
2630 RETURN
```

PROGRAM LISTING FOR VELOCITY-FUNCTION NORMAL MOVEOUT CORRECTION

```
100 PRINT "L_VELOCITY-FUNCTION NMO CORRECTION OF 12-TRACE RECORD"
110 INIT
120 DIM A$(17),C$(8),D$(19),G$(1),H$(11),J$(18),L$(2),M$(17),N$(7)
130 DIM O$(7),S$(1),U$(9),V$(35),X1(12)
140 REM ** INPUT VEL FUNCTION DATA
150 GOSUB 1730
160 Q1=1
170 M$="NO MUTE APPLIED"
180 PRINT "G_G_G___INSERT NMO MDT IN 4051 AND OBS-DATA MDT IN 4924"
190 PRINT "Name of NMO MDT = ";
200 INPUT N$
210 PRINT "Rec# on NMO MDT = ";
220 INPUT R5
230 F0=R5*12-10
240 PRINT "Name of OBS MDT = ";
250 INPUT O$
260 PRINT "Rec# on OBS MDT = ";
270 INPUT R1
280 F1=R1*12-10
290 PRINT "DO YOU WANT TO APPLY MUTE? (Y OR N) ";
300 INPUT G$
310 IF G$="N" THEN 340
320 DIM J8(12)
330 Q1=2
340 REM ** ENTER OBS DATA SI AND DELAY
350 GOSUB 520
360 REM ** ENTER SPREAD DATA AND MUTE-START TIMES
370 GOSUB 1860
380 IF Q1=1 THEN 410
390 PRINT "  Cos-taper time = ";
400 INPUT C1
410 REM ** ESTABLISH NMO TIME RANGE
420 GOSUB 600
430 REM ** TEST DATA LIMITS
440 GOSUB 790
450 REM ** SET COS TAPER & COMPUTE MUTE COEFF
460 IF Q1=1 THEN 480
470 GOSUB 920
480 REM ** COMPUTE, APPLY, AND STORE
490 GOSUB 1020
500 PRINT "G_G_G_PROGRAM COMPLETED"
510 END
520 REM ** SUB: ENTER OBS DATA S.I. AND DELAY TIME
530 PRINT "__  OBS MDT S.I. = ";
540 INPUT S1
550 S3=S2/S1
560 PRINT "  OBS MDT delay = ";
570 INPUT L
580 J7=L/S1+1
590 RETURN
600 REM ** SUB: NMO TIME RANGE
610 PRINT "__FOR NMO-CORRECTED RECORD:"
620 PRINT "Start NMO time = ";
630 INPUT T2
640 IF T2=>T(1) THEN 670
650 PRINT "G_G_G_ERROR! NMO START TIME < FIRST TIME OF VEL FUNCTION"
660 GO TO 620
670 J2=T2/S1+1
680 PRINT "  End NMO time = ";
690 INPUT T3
700 IF T3<=T(N) THEN 730
710 PRINT "G_G_G_ERROR! NMO END TIME > LAST TIME OF VEL FUNCTION"
720 GO TO 680
730 J3=T3/S1+1
740 N2=J3-J2+1
750 N3=(N2-1)/S3+1
760 N4=3*N3
770 REM ** REORDER VELOCITY FUNCTION
780 GOSUB 2790
790 REM ** SUB: TEST DATA LIMITS
```

```
800 CALL "MIN",X1,X4,I4
810 CALL "MAX",X1,X3,I3
820 T4=SQR(X4/V(1)*X4/V(1)+T2*T2)
830 IF T4=>L THEN 860
840 PRINT "G_G_G_ERROR: INITIAL COINCIDENT RAY TIME TOO SMALL"
850 GO TO 620
860 J4=INT(0.5+(T4-L)/S1)+1
870 T4=(J4-1)*S1
880 T5=SQR(X3/V(N)*X3/V(N)+T3*T3)
890 J5=INT(0.5+(T5-L)/S1)+1
900 T5=(J5-1)*S1
910 RETURN
920 REM ** SUB: SET COS TAPER
930 C$=STR(C1)
940 M$=C$&" msec cosine"
950 C1=INT(C1/S1+0.5)
960 DIM K9(C1)
970 C2=PI/C1
980 FOR N=1 TO C1
990 K9(N)=0.5-0.5*COS(C2*N)
1000 NEXT N
1010 RETURN
1020 REM ** SUB: COMPUTE,APPLY,AND STORE
1030 FOR J=1 TO 12
1040 DELETE D$,O
1050 J9=1001 MAX J5
1060 DIM D$(3005),O(J9)
1070 IF J>1 THEN 1110
1080 FIND @2:F1
1090 READ @2:H$,D$
1100 GO TO 1130
1110 FIND @2:F1+J-1
1120 READ @2:D$
1130 O=511
1140 CALL "HEXDEC",D$,O,LEN(D$),3
1150 DELETE D$
1160 O=O-511
1170 IF Q1=1 THEN 1260
1180 FOR K=1 TO J8(J)
1190 O(K)=0
1200 NEXT K
1210 P=0
1220 FOR N=K TO K+C1-1
1230 P=P+1
1240 O(N)=K9(P)*O(N)
1250 NEXT N
1260 GOSUB 1310
1270 GOSUB 1480
1280 GOSUB 1580
1290 NEXT J
1300 RETURN
1310 REM ** SUB: COMPUTE NMO TIMES AS INDICES
1320 DIM T(N3)
1330 J6=J2-S3
1340 T6=T2-S2
1350 FOR I=1 TO N3
1360 J6=J6+S3
1370 T6=T6+S2
1380 T(I)=T6
1390 NEXT I
1400 DIM M(N3)
1410 FOR K=1 TO N3
1420 M(K)=SQR(T(K)*T(K)+X1(J)/V(K)*X1(J)/V(K))-T(K)
1430 M(K)=INT(0.5+M(K)/S1)
1440 NEXT K
1450 M=M-J7
1460 DELETE T
1470 RETURN
1480 REM ** SUB: MAKE NMO CORRECTIONS
1490 DIM D(N3)
```

```
1500 D=0
1510 J6=J2-S3
1520 FOR I=1 TO N3
1530 J6=J6+S3
1540 D(I)=O(J6+M(I))
1550 NEXT I
1560 DELETE M,O
1570 RETURN
1580 REM ** SUB: STORE NMO-CORRECTED DATA
1590 DELETE D$
1600 DIM D$(N4)
1610 D$=" "
1620 D=D+511
1630 CALL "DECHEX",D$,D,N3,3
1640 DELETE D
1650 IF J>1 THEN 1690
1660 FIND F0
1670 WRITE H$,M$,N$,O$,V$,N3,R1,S2,T2,T3,D$
1680 GO TO 1710
1690 FIND F0+J-1
1700 WRITE D$
1710 CLOSE
1720 RETURN
1730 PRINT "G_G_G___INSERT VEL-FUNCTION MDT IN 4924";
1740 PRINT "   FILE NO = ";
1750 INPUT F2
1760 FIND @2:F2
1770 READ @2:N
1780 DIM T(N),V1(N)
1790 READ @2:A$,D$,U$,V$,T,V1
1800 PRINT "__VELOCITY FUNCTION USING ";V$
1810 PRINT A$;"  ";U$;"  ";D$
1820 S2=(T(N)-T(1))/(N-1)
1830 PRINT "FUNCTION BEGINS AT T=";T(1);" AND ENDS AT ";T(N);" ";
1840 PRINT "WITH A SI=";S2
1850 RETURN
1860 REM ** SUB: SPREAD DATA & MUTE START TIME
1870 PRINT "__SPREAD TYPES"
1880 PRINT "   1. Single-ended, equally spaced detectors"
1890 PRINT "   2. Split, equally spaced detectors"
1900 PRINT "   3. Spread with unequally spaced detectors"
1910 PRINT "ENTER SELECTION FROM ABOVE LIST; NUMBER = ";
1920 INPUT N0
1930 GO TO N0 OF 1940,1960,1980
1940 GOSUB 2000
1950 GO TO 1990
1960 GOSUB 2230
1970 GO TO 1990
1980 GOSUB 2640
1990 RETURN
2000 REM ** SUB: SINGLE-ENDED SPREAD
2010 PRINT "__FOR THE SINGLE-ENDED SPREAD:"
2020 PRINT " Trace  1 offset = ";
2030 INPUT X1(1)
2040 IF Q1=1 THEN 2070
2050 PRINT "Tr. 1 mute start = ";
2060 INPUT T8
2070 PRINT " Trace 12 offset = ";
2080 INPUT X1(12)
2090 IF Q1=1 THEN 2120
2100 PRINT "Tr. 12 mute start = ";
2110 INPUT T9
2120 X2=(X1(12)-X1(1))/11
2130 IF Q1=1 THEN 2190
2140 T7=(T9-T8)/11
2150 FOR K=1 TO 12
2160 T0=(K-1)*T7+T8+L
2170 J8(K)=INT(T0/S1+1.5)
2180 NEXT K
2190 FOR J=2 TO 11
2200 X1(J)=X1(J-1)+X2
```

```
2210 NEXT J
2220 RETURN
2230 REM ** SUB: SPLIT SPREAD
2240 PRINT "__FOR THE SPLIT SPREAD:"
2250 PRINT " Trace 1 offset = ";
2260 INPUT X1(1)
2270 IF Q1=1 THEN 2300
2280 PRINT "Tr. 1 mute start = ";
2290 INPUT T8
2300 PRINT " Trace 6 offset = ";
2310 INPUT X1(6)
2320 IF Q1=1 THEN 2400
2330 PRINT "Tr. 6 mute start = ";
2340 INPUT T9
2350 T7=(T9-T8)/5
2360 FOR K=1 TO 6
2370 T0=(K-1)*T7+T8+L
2380 J8(K)=INT(T0/S1+1. 5)
2390 NEXT K
2400 X2=(X1(6)-X1(1))/5
2410 FOR J=2 TO 5
2420 X1(J)=X1(J-1)+X2
2430 NEXT J
2440 PRINT " Trace 7 offset = ";
2450 INPUT X1(7)
2460 IF Q1=1 THEN 2490
2470 PRINT "Tr. 7 mute start = ";
2480 INPUT T8
2490 PRINT "Trace 12 offset = ";
2500 INPUT X1(12)
2510 IF Q1=1 THEN 2590
2520 PRINT "Tr. 12 mute start= ";
2530 INPUT T9
2540 T7=(T9-T8)/5
2550 FOR K=7 TO 12
2560 T0=(K-7)*T7+T8+L
2570 J8(K)=INT(T0/S1+1. 5)
2580 NEXT K
2590 X3=(X1(12)-X1(7))/5
2600 FOR J=8 TO 11
2610 X1(J)=X1(J-1)+X3
2620 NEXT J
2630 RETURN
2640 REM ** SUB: UNEQUALLY SPACED SPREAD
2650 PRINT "FOR THE UNEQUALLY SPACED SPREAD:"
2660 IF Q1=1 THEN 2690
2670 PRINT "TRACE NO.   OFFSET   MUTE START TIME"
2680 GO TO 2700
2690 PRINT "TRACE NO.   OFFSET"
2700 FOR J=1 TO 12
2710 PRINT "    "; J; "            ";
2720 INPUT X1(J)
2730 IF Q1=1 THEN 2770
2740 PRINT "K_                          ";
2750 INPUT T0
2760 J8(J)=INT(T0/S1+1. 5)
2770 NEXT J
2780 RETURN
2790 REM ** SUB: REORDER VEL FUNCTION
2800 DIM V(N3)
2810 IF T2=T(1) THEN 2880
2820 K=(T2-T(1))/S2
2830 FOR J=1 TO N3
2840 K=K+1
2850 V(J)=V1(K)
2860 NEXT J
2870 GO TO 2890
2880 V=V1
2890 DELETE V1
2900 RETURN
```